

Práctica 2 - Sistemas Web

Memoria

Complucine [FDI-Cines]

15 de abril de 2021

Índice

1. Introducción	4
2. Listado de Scripts para las Vistas	5
2.1. Pantalla de inicio	5
2.2. Inicio de Sesión / Registro	5
2.3. Cartelera	6
2.4. Selección de Cine	6
2.5. Selección de Horario	6
2.6. Pantalla de Pago	6
2.7. Quiénes Somos	7
2.7.1. Sobre FDI-Cines	7
2.7.2. Destalles	7
2.7.3. Bocetos	7
2.7.4. Miembros	7
2.7.5. Planificación	7
2.8. Paneles	7
2.8.1. Panel de Usuario	7
2.8.2. Panel de Gerente	7
2.8.3. Panel de Administrador	8
2.9. Términos y Condiciones	8
2.10. Contacto	8
3. Lista de Scripts Adiciones	9
3.1. DAO	9
3.2. Inicio de sesión y Registro de usuarios	9
3.2.1. User DAO	9
3.2.2. User DTO	9
3.2.3. Validación de inicio de sesión	9
3.2.4. Validación de registro	9
3.3. Panel de Administrador	10
3.3.1. Film DAO	10
3.3.2. Film DTO	10
3.3.3. Añadir/Editar/Eliminar Películas	10
3.4. Panel de Gerente	10
3.4.1. Sesiones DAO	11
3.4.2. Sesiones DTO	11
3.4.3. Añadir/Editar/Eliminar Salas	11
3.5. Panel de usuario	11
3.5.1. Historial de Compras	11
3.5.2. Cambiar Contraseña	12
3.5.3. Datos de Pago	12

3.5.4. Eliminar cuenta	12
4. Estructura de la Base de Datos	13
4.1. Modelo Entidad-Relación	13
4.2. Estructura de la Base de Datos	14
4.3. Tablas y Relaciones	14
4.3.1. Cinema (Cine)	14
4.3.2. Hall (Sala)	14
4.3.3. Seat (Butaca)	14
4.3.4. Film (película)	15
4.3.5. Session (Sesión)	15
4.3.6. Purchase (Compra)	15
4.3.7. User (usuario)	16
4.3.8. Admin (Administrador)	16
4.3.9. Manager (Gerente)	16
4.3.10. Promotion (Promoción)	16
4.4. Scripts PHP	16

1. Introducción

Esta memoria corresponde al trabajo realizado en la práctica 2 de la asignatura de APLICACIONES/SISTEMAS WEB y el **proyecto Complucine del grupo FDI-Cines**.

Se ha intentado seguir una estructura de *Vista/Controlador* para cada una de las pantallas de la web.

Además, se ha estructurado el acceso a la base de datos con un patrón DAO.



Figura 1: Pantalla de inicio de la web de Complucine

2. Listado de Scripts para las Vistas

2.1. Pantalla de inicio

Pantalla inicial, se cargará por defecto al entrar en la web.

Esta pantalla incluye poca información y de forma muy visual, con la intención de que el usuario encuentre una web amigable, sencilla e intuitiva.

La información más destacada que podremos ver en este script es:

- Últimos estrenos: las portadas de las últimas 4 películas añadidas a la Base de Datos.
- Película destacada: se mostrará el título y la portada¹ de una película aleatoria de la Base de Datos.
- Promociones: por último, aparecerán algunos descuentos y promociones, para invitar al usuario a realizar la compra de sus entradas en nuestros cines. Estas promociones pueden ser, desde descuentos directos en el precio final de su compra, hasta regalos para obtener *snacks* antes de entrar a la película.

2.2. Inicio de Sesión / Registro

En esta pantalla los usuarios que estén navegando por la web, podrán iniciar sesión (si ya tenían una cuenta registrada en nuestra Base de Datos), o por el contrario, registrarse (en caso de que no lo hubieran hecho previamente).

En esta vista, **por defecto aparecerá el formulario para iniciar sesión** (pues suele ser la acción más común).

Pero a la vez, aparecerá un botón llamado REGISTRARSE, para cambiar a la vista del formulario de registro si el usuario así lo desea. Una vez cambiada a esta vista el formulario de **inicio de sesión** será remplazado por el formulario de **registro** y el botón para cambiar de formulario, ahora volvería a mostrar el formulario de inicio de sesión en caso de ser pulsado.

¹Pulsando en cualquiera de las portadas, tanto de los últimos estrenos, como de las películas destacadas, te llevará a su correspondiente sección dentro de la pantalla de **cartelera**.

2.3. Cartelera

Esta pantalla incluye información sobre todas las películas disponibles en nuestros cines.

Se hará una lectura de la tabla correspondiente a las películas en la Base de Datos y se irá generando un recuadro para mostrar una a una todas las películas. Desde esta pantalla el flujo de compra sería:

Selección de Película >> Selección de Cine >> Selección de Horario
>> Aplicar promociones (opcional >> Realizar Compra

2.4. Selección de Cine

Esta pantalla mostrará un mapa o lista (dependiendo de cómo avance la práctica y lo complicado que pueda llegar a ser lo primero), con todos los cines y su localización física.

De esta forma el usuario podrá escoger aquel cine que más le convenga. Y a partir de ahí seleccionar la película.

Desde esta pantalla el flujo de compra sería:

Selección de Cine >> Selección de Película >> Selección de Horario
>> Realizar Compra y aplicar promociones (opcional)

2.5. Selección de Horario

En esta pantalla se escoge el horario al que se quiere acceder a la sala para ver la película.

La sala en la que el cliente verá la película, se escoge en función de la película, cine y horario escogido previamente.

A esta pantalla sólo se puede llegar habiendo escogido previamente una película y un cine (independientemente del orden de elección).

2.6. Pantalla de Pago

En esta pantalla se procederá a confirmar las selecciones que el usuario ha hecho previamente y realizar el pago, De forma opcional, el usuario podrá aplicar promociones y/o descuentos a su compra.

2.7. Quiénes Somos

Esta sección de pantallas adicionales, contiene simplemente las pantallas de la *práctica 1*, hemos querido mantenerlas, pues muestran información relevante sobre este proyecto y los integrantes del grupo.

2.7.1. Sobre FDI-Cines

Pequeña descripción del proyecto y nuestro grupo.

2.7.2. Destalles

Información un poco más detallada del objetivo de esta aplicación web.

2.7.3. Bocetos

Bocetos con las pantallas que, en un principio, debería tener la aplicación final.

2.7.4. Miembros

Información sobre los componentes del equipo.

2.7.5. Planificación

Planificación estimada, con hitos, objetivos y fechas de entrega aproximadas.

2.8. Paneles

En esta sección se procede a detallar los diferentes paneles que existen, en función del tipo de usuario que haya iniciado sesión.

Cada panel es un script diferente y un usuario no puede acceder a un panel que no le corresponda, sólo al suyo, según su tipo de usuario.

2.8.1. Panel de Usuario

Panel en el que el usuario podrá ver y modificar la información sobre su cuenta. Además de ver sus compras y promociones disponibles.

2.8.2. Panel de Gerente

Panel accesible únicamente para los *Gerentes* de un cine. En el podrán **añadir, editar y eliminar** tanto **salas como sesiones** del cine que el Gerente tenga asignado.

2.8.3. Panel de Administrador

Panel accesible únicamente para los *Administradores* de la aplicación. En el podrán **añadir, editar y eliminar cines, películas, promociones, gerentes y otros administradores.**

Adicionalmente los administradores podrán usar una funcionalidad para *ver como* un rol específico. Estos roles podrán ser *usuario no registrado, usuario registrado o gerente*. Dando así la función al administrador de comprobar que lo que otros roles visualizan en la aplicación es correcto.

2.9. Términos y Condiciones

Pantalla con los términos y condiciones de la aplicación, que todo usuario debe aceptar antes de registrarse en la misma.

Como nadie se los lee, no son muy relevantes.

2.10. Contacto

Pantalla en la que un usuario podrán enviar un mensaje a un Administrador o Gerente, en función de un problema duda que le haya surgido al usar alguna funcionalidad de la aplicación.

3. Lista de Scripts Adiciones

3.1. DAO

Esta clase es la encargada de hacer la conexión segura a la Base de Datos. De ella heredan todas las clases DAO que quieran realizar algún tipo de consulta o modificación en la base de datos.

3.2. Inicio de sesión y Registro de usuarios

Estos son los scripts adicionales relacionados con el inicio de sesión y registro de los usuarios.

3.2.1. User DAO

Esta clase es la encargada de hacer las consultas e inserciones a la base de datos, relacionadas únicamente y exclusivamente con la tabla de usuarios.

Hereda de la clase DAO, para hacer la conexión segura a la base de datos.

3.2.2. User DTO

Esta clase sirve para crear una entidad (un objeto) usuario, con los que poder asignar y tratar los atributos, sin modificar información en la base de datos.

3.2.3. Validación de inicio de sesión

En este script se comprueban los datos introducidos por el usuario en el formulario de **inicio de sesión**. De tal forma que nos aseguremos que cada campo que recibimos, corresponde con lo esperado y evitando problemas de seguridad.

Una vez validados los datos, se hacen las comprobaciones para saber si el usuario existe en la base de datos. En caso de éxito, asigna los atributos de *\$_SESSION* correspondientes. En caso contrario, devuelve un mensaje de error.

3.2.4. Validación de registro

En este script se comprueban los datos introducidos por el usuario en el formulario de **registro**. De tal forma que nos aseguremos que cada campo que recibimos, corresponde con lo esperado y evitando problemas de seguridad.

Una vez validados los datos, se hacen las comprobaciones para saber si el usuario existe en la base de datos. En caso de que no exista, se procede a hacer la inserción en la base de datos y devolver un mensaje de éxito en la operación. En caso contrario, devuelve un mensaje de error.

3.3. Panel de Administrador

Estos son los scripts adicionales del *Panel de Administrador* que se han implementado en esta práctica. En la siguiente práctica, se añadirán nuevas funcionalidades progresivamente.

3.3.1. Film DAO

La clase film DAO se encarga de realizar todas la consultas a la base de datos relacionadas con películas. Hereda de la clase DAO, que se encarga de hacer la conexión a la base de datos.

3.3.2. Film DTO

La clase film DAO se encarga de realizar todas la consultas a la base de datos relacionadas con películas. Hereda de la clase DAO, que se encarga de hacer la conexión a la base de datos.

3.3.3. Añadir/Editar/Eliminar Películas

Este script se encarga de tres funciones a escoger por el usuario:

- **Añadir:** El script recoge los datos de la película comprueba si se adaptan al campo al que corresponden , comprueba si no se encuentra ya en la base de datos y posteriormente la añade a la base de datos, si la película ya se encuentra en la base de datos o algún campo es vacío se retorna un mensaje de error.
- **Editar:** Muy similar a añadir la única diferencia es que en este caso se comprueba si existe en la base de datos con el id y en caso afirmativo se añade a la base de datos e igual que en el caso anterior si algún campo no es el esperado o la película no se encuentra en la base de datos se retorna un mensaje de error.
- **Eliminar:** En este caso solo se comprueba a través del id que exista en la base de datos la película y posteriormente se borra si existe, en caso contrario devuelve un mensaje de error.

3.4. Panel de Gerente

Estos son los scripts adicionales del *Panel de Gerente* que se han implementado en esta práctica. En la siguiente práctica, se añadirán nuevas funcionalidades progresivamente.

3.4.1. Sesiones DAO

La clase *SessionDAO* se encarga de realizar todas las consultas a la base de datos relacionadas con las sesiones como editar, eliminar, seleccionar o crear una sesión. Hereda de la clase DAO, que se encarga de hacer la conexión a la base de datos.

3.4.2. Sesiones DTO

La clase *SessionDTO* crea un objeto que almacena todos los atributos que hay en la base de datos de una sesión y poder tratar estos de forma mas segura y clara.

3.4.3. Añadir/Editar/Eliminar Salas

Este script se encarga de tres funciones a escoger por el gerente:

- **Añadir:** Aquí recoge todos los datos mediante un formulario que se lo envía a *validateSession.php*, en este php comprueba si están los campos bien puestos y si existe una sesión igual, en caso de que todo este bien se añade a la base de datos creando *SessionDAO* y llamando a *createSession(...)*. En caso de que de error retorna un mensaje de error.
- **Editar:** A la hora de editar el gerente mediante un formulario no puede ni editar el id ni el cine donde se encuentra, a la hora de seleccionar tanto salas como películas solo puede elegir aquellas que existen y puede cambiar a su gusto tanto el horario(día y hora), precio y formato, también tiene una opción extra donde puede repetir esta sesión durante x días. Pasa el formulario a *validateSession.php* comprueba de que este todo bien y lo cambia en la base de datos gracias a *SessionDAO*.
- **Eliminar:** En el caso de que el gerente quiera eliminar una sesión, a través del mismo formulario de editar se envía con el botón de eliminar donde cambia la acción en *validateSession.php*, se comprueba los campos de este y se elimina de la base de datos.

3.5. Panel de usuario

Los siguientes scripts no se han desarrollado para la práctica 2, aun así se listan a continuación.

3.5.1. Historial de Compras

Aquí el cliente puede consultar su historial de compras, se verán datos como la fecha del pedido, la película, y el precio de la compra.

3.5.2. Cambiar Contraseña

Aquí el cliente puede, introduciendo su contraseña actual (por motivos de seguridad), cambiar la contraseña por otra nueva.

3.5.3. Datos de Pago

Aquí el cliente puede consultar los datos de pago que tiene guardados, en caso de que haya alguno.

3.5.4. Eliminar cuenta

Aquí el cliente puede, introduciendo su contraseña actual (por motivos de seguridad), borrar permanentemente su usuario.

4. Estructura de la Base de Datos

4.1. Modelo Entidad-Relación

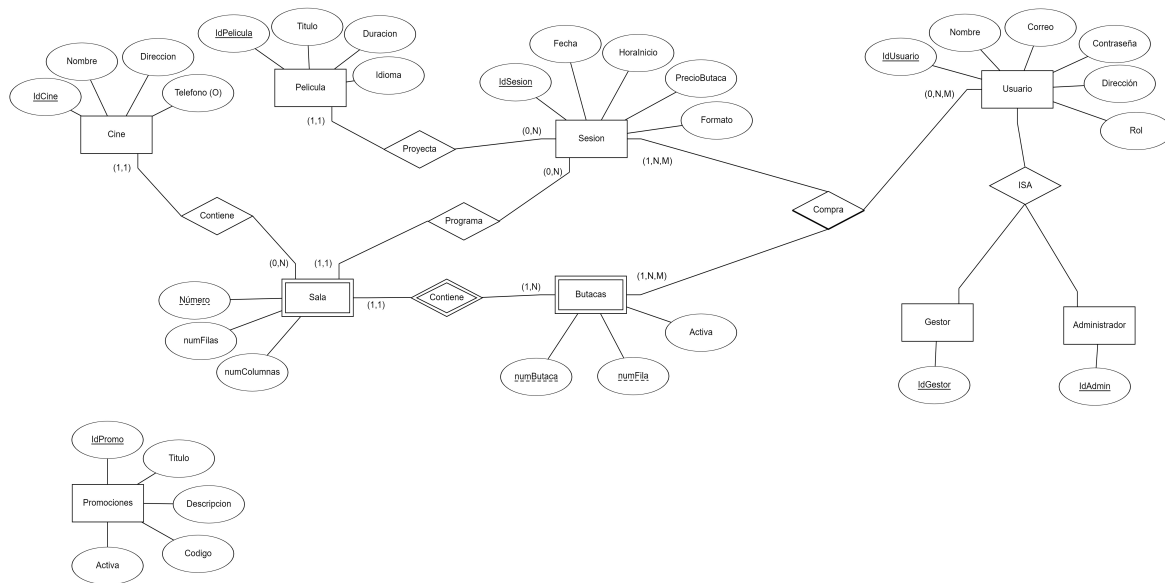


Figura 2: Modelo Entidad-Relación de la Base de Datos de Complucine

4.2. Estructura de la Base de Datos

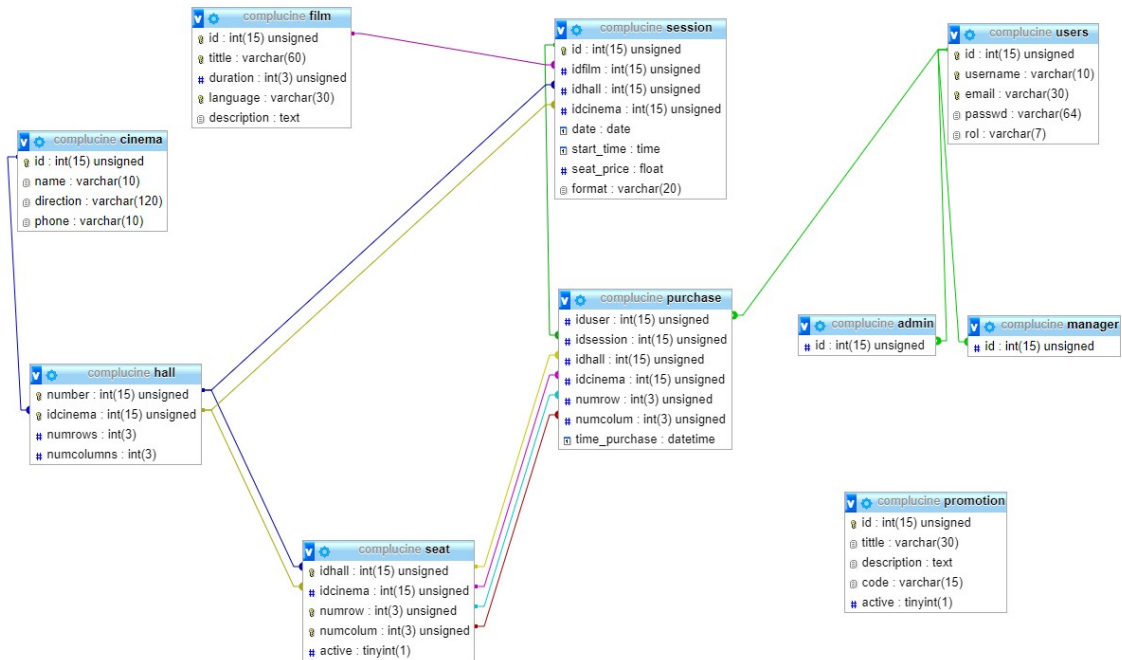


Figura 3: Estructura de la Base de Datos de CompluCine

4.3. Tablas y Relaciones

4.3.1. Cinema (Cine)

Tabla derivada de la entidad cine, la cual está identificada por un id que es único. Contiene los atributos necesarios para almacenar la información requerida: *nombre*, *dirección* y *teléfono*.

4.3.2. Hall (Sala)

Tabla derivada de la entidad sala, la cual es una entidad débil ya que su clave candidata, *number*, no permite identificarla inequívocamente, pues pueden existir salas con el mismo valor para el atributo *number* en distintos cines.

Además de los atributos necesarios para almacenar su información (*number*, *numRows*, *numColumns*), contiene un atributo *idCinema*, el cual forma parte de la clave foránea de esta tabla referenciando a la tabla cinema.

4.3.3. Seat (Butaca)

Tabla derivada de la entidad butaca, la cual es una entidad débil ya que su clave candidata, *numRow* y *numColum*, no permite identificarla inequívocamente, pues pueden

existir butacas con el mismo valor para estos atributos en distintas salas y cines. Además de los atributos necesarios para almacenar su información (*numRow*, *numColumn*, *active*), contiene los atributos *idCinema* e *idHall*, los cuales forman parte de la clave foránea de esta tabla referenciando a la tabla HALL.

4.3.4. Film (película)

Tabla derivada de la entidad película, la cual está identificada por un id que es único. Contiene los atributos necesarios para almacenar la información requerida: título, duración, idioma y descripción.

Además se incluye una restricción para asegurar que no se puedan añadir películas con el mismo título e idioma. Elegimos esta implementación en lugar de hacer que el atributo sea multivalorado ya que en ocasiones el título de una misma película puede variar en función del idioma, por ello a efectos prácticos tendremos una nueva película en nuestra base de datos por cada idioma, que a su vez facilita su selección a la hora de crear una nueva sesión.

4.3.5. Session (Sesión)

Tabla derivada de la entidad sesión, la cual está identificada por un id que es único. Decidimos decantarnos por esta implementación para poder almacenar más de una sesión asociada tanto a una sala como a una película.

Además de los atributos necesarios para almacenar la información de la sesión (día, hora de inicio, precio de la butaca y formato), contiene los atributos *idCinema* e *idHall*, los cuales forman parte de la clave foránea de esta tabla referenciando a HALL y el atributo *idFilm* el cual forma parte de la clave foránea referenciando a FILM. Añadimos también una restricción para que el gerente no pueda añadir una misma sesión en un mismo día y con una misma hora de inicio.

4.3.6. Purchase (Compra)

Tabla derivada de la relación ternaria ente las entidades *sesión*, *butaca* y *usuario*. Todos los atributos que contiene esta tabla forman tres claves foráneas:

- **idUser** el cual referencia a la tabla User.
- **idSession** el cual referencia a la tabla Session.
- **idCinema**, **idHall**, **numRow**, **numColumn** los cuales referencian a la tabla Seat.

Como excepción tenemos el atributo de la relación *time_purchase* que nos sirve para almacenar la fecha y hora de la compra.

4.3.7. User (usuario)

Tabla derivada de la entidad usuario, la cual está identificada por un id que es único. Contiene los atributos necesarios para almacenar la información requerida: *username*, *email*, *passwd* y *rol*².

4.3.8. Admin (Administrador)

Tabla derivada de la relación ISA con la tabla *User*. Únicamente contiene el id que referencia a Usuario.

4.3.9. Manager (Gerente)

Tabla derivada de la relación ISA con la tabla *User*. Únicamente contiene el id que referencia a Usuario.

4.3.10. Promotion (Promoción)

Tabla derivada de la entidad promoción, la cual está identificada por un id que es único. Contiene los atributos necesarios para almacenar la información requerida: *tittle*, *description*, *code*, *active*.

4.4. Scripts PHP

En nuestra aplicación para realizar la gestión de acceso a la BD y realizar consultas o modificaciones, hemos implementado la clase DAO (script *dao.php*) la cual se encarga de establecer la conexión con la BD y posterior desconexión.

A partir de este punto hemos creado nuevas clases que heredan de DAO y se encarga de realizar las consultas o modificaciones de cada tabla y devuelven los resultados de estas.

En esta versión hemos implementado:

- **user_dao.php**: se encargar de realizar consultas y modificaciones sobre la tabla *user* y devolver el resultado de estas.
- **fil_dao.php**: se encargar de realizar consultas y modificaciones sobre la tabla *film* y devolver el resultado de estas.
- **session_dao.php**: se encargar de realizar consultas y modificaciones sobre la tabla *session* y devolver el resultado de esta.
- **hall_dao.php** se encargar de realizar consultas y modificaciones sobre la tabla *hall* y devolver el resultado de esta.

²El atributo *rol* desaparecerá en posteriores versiones, era necesario cuando no teníamos implementada la relación ISA con *admin* y *manager*.

Esta obra está bajo una licencia Creative Commons “Reconocimiento-NoCommercial-CompartirIgual 4.0 Internacional”.

